

METHOD AND APPARATUS FOR WIRELESS NETWORK EXPANSION

FIELD OF THE INVENTION

The present invention relates to the field of wireless networks. In one form the invention relates to the management of a wireless network and the mobile network devices therein. Hereinafter, mobile or wireless network devices are referred to as devices for convenience.

It will also be convenient to hereinafter describe the invention in relation to management of a wireless network with the use of algorithms and protocols for forming a wireless network dynamically, detection of the presence of a network, synchronizing to the network and access techniques, each of which may be applied to mobile devices which in turn may operate in a network environment. However, it should be appreciated that the present invention is not limited to that use, only.

BACKGROUND OF THE INVENTION

Many wireless network protocols rely on Time Division Multiple Access (TDMA) techniques for communications between different wireless devices. Examples are GSM, Bluetooth and DECT (Digital European Cordless Telecommunication) specifications. In these methods, each device is allocated a fixed transmission timeslot. For convenience, transmission timeslot is referred to hereinafter simply as a timeslot. The device may transmit and receive data in its given timeslot. For example, in DECT, a 1-second time frame is divided into 24 timeslots, of which, 12 are used for transmission and 12 are for reception. A device with a DECT protocol is assigned a fixed timeslot, and it may transmit data in its assigned timeslot.

To synchronize and allocate timeslots, most networks have a master that transmits a beacon and allocates resources over the available airtime. The master may be pre-assigned and all communication accordingly may go through the master. In such an arrangement two slave devices may not communicate to each other directly, in which case, the information is first relayed to the master.

In contrast, Bluetooth does not pre-assign a fixed master. In this instance, the device which initiates the communication becomes the master. Any other new devices which enter the network may then become the slave in this piconet. There may only be one master in a piconet. However, a
5 problem with timing synchronization may arise when 2 piconets come within the vicinity of each other (scatternets result). This situation may be particularly acute when a particular device happens to be a slave member in two separate piconets. The slave may have to keep two sets of timing information and switch between the two timings according to the particular
10 network that it is currently active in.

Any discussion of documents, devices, acts or knowledge in this specification is included to explain the context of the invention. It should not be taken as an admission that any of the material forms a part of the prior art base or the common general knowledge in the relevant art in Singapore or
15 elsewhere on or before the priority date of the disclosure and claims herein.

An object of the present invention is to provide a useful method of managing the operation of a wireless network.

A further object of the present invention is to alleviate at least one disadvantage that may be associated with the prior art.
20

SUMMARY OF THE INVENTION

In one aspect, the present invention provides a data transmission packet format for use by a first device communicating in a communication network, the data transmission packet format defined by fields comprising
25 source device ID for identifying the first device as a transmission source; timeslot for identifying the timeslot number currently used by the first device; neighborhood list for identifying whether at least one other timeslot in the vicinity of the first device has been occupied for transmission by a second device and so as to identify whether the at least one other timeslot is available
30 for a receiving third device.

In another aspect the present invention provides a method of data encoding for data packets in a communication network, the method

comprising the steps of enclosing a data byte with a start bit and a stop bit; wherein the start bit comprises a first predetermined logic level of a first predetermined duration followed by a transition to a complimentary second predetermined logic level of a second predetermined duration, and wherein
5 the stop bit comprises a third predetermined logic level of a third predetermined duration followed by a transition to a complimentary fourth predetermined logic level of a fourth predetermined duration.

The data encoding method may further comprising the steps of sending a start bit prior to sending the first bit of a payload data byte; sending a stop
10 bit after sending the last bit of a payload data byte.

In another aspect the present invention provides a method of synchronizing devices for communicating in a network, the method comprising the steps of detecting a start symbol at a receiving device by timing the duration of a logic signal level on a received data transmission, changing the
15 state of the receiving device upon detecting the start symbol, to a wait state wherein the receiving device waits to receive the start of a payload data transmission in the form of a predetermined logic transition, adjusting the internal clock of the receiving device immediately upon detecting the predetermined logic transition, so as to synchronize the receiving device with
20 other devices receiving the data transmission within the network.

The present invention also provides a method of dynamically allocating a timeslot for a receiving device in a communication network, the method comprising the steps of receiving a data transmission from the network; copying a received timeslot from the data transmission; copying a received
25 neighborhood list from the data transmission; searching the received neighborhood list for an unused timeslot in accordance with a predetermined logical search strategy so as to locate and allocate a timeslot to the receiving device.

In yet another aspect the present invention provides a method of
30 providing network access for a device adapted to receive data, the method comprising the steps of receiving a data packet formatted as described herein; processing the received data packet as follows:

copying the received neighborhood list;

copying the received timeslot;

forming a network with the device and the transmitting device if the device does not have any neighbor and is not in network;

5 determine a dynamic allocation of a timeslot in accordance with a method as described herein based on the following predetermined criteria:

a) if the device is not in network and its ID value is less than the transmitting device's ID, perform a dynamic allocation of a timeslot;

b) if the device is not in network and the transmitting device is in
10 network, perform a dynamic allocation of a timeslot;

c) if the device and the transmitting device are already in a network, do not perform a dynamic allocation of a timeslot.

In another aspect the present invention provides a protocol for dynamically expanding a network of communicating devices, the protocol
15 comprising the following states of operation:

sleep, wherein a device's transmitter and receiver are inactive and the device is waiting for an external trigger;

wake up, wherein a device's transmitter and receiver are enabled by an
external trigger;

20 scan, wherein a device's receiver is continuously receiving such that it scans for transmissions from other devices;

listen, wherein a device is allocated a timeslot, synchronized to the timeslots of other devices, and such that the device only has its receiver operating during its allocated timeslot;

25 normal, wherein a device is allocated a timeslot without being synchronized to the timeslots of other devices and, such that the device transmits during its allocated timeslot and receives during every other timeslot;

network, wherein a device is allocated a timeslot for receiving and for transmitting and synchronized to the timeslots of other devices in the network.

30 The present invention also provides apparatus adapted to format a data packet, encode data, synchronize devices, dynamically allocate a timeslot, provide network access, or provide a protocol for dynamically

expanding a network, as disclosed herein, said apparatus including processor means adapted to operate in accordance with a predetermined instruction set, said apparatus, in conjunction with said instruction set, being adapted to perform the methods as disclosed herein.

5 Furthermore the present invention provides a computer program product including a computer usable medium having computer readable program code and computer readable system code embodied on said medium for any one or more of formatting a data packet, encoding data, synchronizing devices, dynamically allocating a timeslot, providing network
10 access, or providing a protocol for dynamically expanding a network, within a data processing system, said computer program product including computer readable code within said computer usable medium for performing the methods as disclosed herein

Other aspects and preferred aspects are disclosed in the specification
15 and / or defined in the appended claims, forming a part of the description of the invention.

In essence, the present invention stems from the realization that providing information in relation to the accessibility of timeslots and allocating use of "free" timeslots to devices in a network by providing this information as
20 part of the network infrastructure, the restrictions of fixed numbers of devices gaining access to a network and its resources may be overcome.

The present invention has been found to result in a number of advantages, such as allowing devices to synchronize automatically without having to keep track of different timings. The present invention also allows
25 two different piconets to merge into one network.

Further scope of applicability of the present invention will become apparent from the detailed description given hereinafter. However, it should be understood that the detailed description and specific examples, while indicating preferred embodiments of the invention, are given by way of
30 illustration only, since various changes and modifications within the spirit and scope of the invention will become apparent to those skilled in the art from this detailed description.

BRIEF DESCRIPTION OF THE DRAWINGS

Further disclosure, objects, advantages and aspects of the present application may be better understood by those skilled in the relevant art by reference to the following description of preferred embodiments taken in conjunction with the accompanying drawings, which are given by way of illustration only, and thus are not limitative of the present invention, and in which:

Figure 1 illustrates a data transmission format in accordance with a first preferred embodiment of the invention.

10 Figure 2 illustrates a data encoding format in accordance with second preferred embodiment of the invention.

Figure 3 shows a typical scheme of a timeslot for transmission.

Figure 4 shows a flowchart for a synchronization algorithm in accordance with a third embodiment of the invention.

15 Figure 5 is an illustration of a Normal mode of a device transmitting in accordance with an embodiment of the invention.

Figure 6 illustrates a Listen mode of a device in accordance with an embodiment of the invention.

20 Figure 7 is a flow chart for an algorithm carrying out a method of adjusting a timeslot in accordance with a further embodiment of the present invention.

Figure 8 is a state transition diagram for a network protocol in accordance with another embodiment of the invention.

25 DETAILED DESCRIPTION OF THE INVENTION

The techniques described in this document allow devices to synchronize automatically without having to keep track of different timings. The techniques also allow two different piconets to merge into one network.

1. Data Transmission Format.

30 1.1 Data Transmission Format

In this technique, data is transmitted in packets. A typical packet 10 with an exemplary data transmission format in accordance with an

embodiment of the invention is illustrated in Figure 1. Each packet consists of the following fields.

- Device Identification ID, 1
- Destination Identification ID, 2
- 5 • Neighbourhood List, 3
- Network List, 4
- Current Timeslot, 5
- Data, 6
- Checksum, 7

10 **Device Identification, ID**

This field 1 identifies the device that is transmitting the data packet 10 at the moment. This field 1 is needed for the protocol to work.

Destination Identification, ID

15 This field 2 identifies the remote device that the current transmitter is sending information to by way of packet 10. This field 2 is not mandatory, but may depend on the application managing the transmission.

Neighbourhood List

20 This field 3 identifies which timeslot in the vicinity of the transmitter has been occupied. The receiver use this information to determine which device is in the vicinity of the transmitter. It also uses this information to assign a timeslot to a receiver which wishes to join an existing network.

Network List

25 This field 4 contains information about the devices in the network that the transmitter is operating in. It gives the receiver information about devices that are operating in the network of the transmitter, but is beyond the reception range of the transmitter.

Current Timeslot

This field 5 identifies the timeslot number that the transmitter is currently using.

30 **Data**

This field 6 comprises the payload or information that the application device is transmitting.

Checksum

This field 7 is used by the receiver to determine if the received packet 10 is valid.

1.2 Data Encoding

5 Data encoding is a method whereby the information bits are coded into symbols so that data may be transmitted. In this algorithm, the bits are coded in such a manner as to enable automatic symbol timing synchronisation and minimize coding redundancy present in, for example, the commonly used Manchester Coding.

10 START and STOP Bits

Reference is made of Figure 2. In this algorithm, each data byte 11 is enclosed by a START BIT 12 and a STOP BIT 13. The START BIT 12 consists of logic Level LOW for half a symbol duration followed by a logic level HIGH for half symbol duration.

15 The STOP Bit 13 consists of logic Level HIGH for half a symbol duration followed by a logic level LOW for half symbol duration.

A START bit 12 will be sent before the first bit of a data byte 11 is sent. After the last bit of the data byte 11 has been sent, a STOP bit 13 will be sent.

START and STOP are used only for the data packets and not for the 20 Preambles 15 and START SYMBOL 14.

The use of START and STOP bits ensure that there will not be long transmissions of consecutive 1s or 0s. It also allows the receiver to synchronize using the transitions.

PREAMBLES

25 Preambles 15 consist of alternating 1's (Logic Level HIGH) and 0's (Logic Level LOW). The number of preambles sent depends on the design of the receiver. The use of the preamble is to condition the receiver for receiving data.

START SYMBOL

30 The START Symbol 14 consists of a 1 for a period of time that exceeds 3 times symbol duration.

It would be well understood by the person skilled in the art that there are other encoding or scrambling techniques that may be employed whilst remaining within the scope of the present invention.

Data Scrambling

- 5 The actual data is scrambled by performing an exclusive OR operation on each data byte with a value of 202 (or CA in hexadecimal form) before adding a START and a STOP bit. Preambles and START symbols are not scrambled.

10 In this algorithm, the Data transmission format and the packet format has been defined in such a way that the START symbol and the Preambles will never appear in any of the data bytes. This provides a means for the receiver to detect that another device is transmitting data and also a means for synchronizing the devices' clock.

Figure 2 Data transmission format.

- 15 1.3 Data Reception and Timing Synchronization Algorithm

20 A time frame is divided into timeslots. An example is shown in Figure 3. Each device transmits in one time slot 8 and receives in the rest of the time slots 9. For example, if the system has 12 timeslots, one time slot 8 would be dedicated to transmission, whilst the rest 9 are for reception. For discussion purposes, let Timeslot #0 be the timeslot for transmission 8. This is an arbitrary number that is given at the start. Assume that each time slot 8, 9 has duration of 100ms, and that there are altogether 12 timeslots.

25 At Timeslot #0, the device would transmit a packet of information as described in Section 2.1. 100ms later, the device would turn on the receiver and listen to the "air" for transmission from other devices. If it receives an information packet, it would process the information and adjusts timing accordingly (See section 2.7 below). Otherwise, it would turn off the receiver and wait for the next time slot.

Detection of data transmission.

- 30 With reference to Figure 4, the receiver detects that another device is transmitting by monitoring the duration of the logic level of the received bit. If

the bit time is longer than a pre-defined period that will not be confused with noise, a carrier is present.

Timing Synchronisation

Every device has an internal clock whose value changes at a fixed time.

- 5 For example, with a 16-bit clock, a device's clock will start from 0 and count up to 65535, and then repeats again. If two devices are synchronized in time, it means that their clock values are the same at the same time.

In the algorithm of Figure 4, the receiver times the duration of Logic level HIGH on the received bit, steps 401, 402 and 403. If the received bit
10 remains at HIGH level for more than two times of a symbol duration step 404, a START Symbol has been detected. The receiver then moves to another state step 405 where it waits for the start of data transmission. The receiver now waits for a negative transition (1 -> 0) step 407 and then a positive transition (0 -> 1) step 408. Immediately after detecting the positive transition
15 step 408, the receiver adjusts its internal clock step 409.

Since each device transmits on a fixed and regular basis, each device would know its exact clock value at the point where it has just completed sending the first START bit. In order to synchronise every device's clock, each receiver only needs to adjust its clock immediately after receiving the first
20 START bit. This will ensure that all devices will be synchronized after receiving a data packet. It would not matter what the receiver's clock value is before detecting the data packet.

1.4 Network protocol

1.4.1 Terms and definitions.

25 Modes

In this algorithm, a device can assume any one of the modes.

- Normal Mode.

With reference to Figure 5, a device is said to be in Normal_Mode if it turns on its transmitter in its own timeslot for transmission and turns on its
30 receiver only in all the other timeslots. This is the normal mode of operation. In the Normal_Mode, the transmitter or receiver is only turned on for a period that is needed for transmission/reception of data. For example, if it takes 12

milliseconds to transmit a packet of data, the transmitter would only be active for the 12 ms to transmit the packet. If the device is not in Normal_Mode, it would not be transmitting.

- Listen Mode.

5 With reference to Figure 6, a device is said to be in Listen_Mode if it does not turn on its transmitter during its allocated timeslot. A device can be operating in the Normal_Mode but in Listen Mode also. This means that during its own time slot for transmission, it would turn on its receiver instead of its transmitter. A device that is not in Normal_Mode but in Listen_Mode would
10 have its receiver turned on all the time. This is also known as the Scanning Mode.

- In Network

A device is said to be in a network if it is able to communicate with at least one device. A device that "sees" no other devices is said to be not in
15 network.

- Neighbours and Neighbourhood

A device is said to be in the neighbourhood of another device if the other device can detect the messages that it is transmitting.

- Neighbourhood List

20 Assuming that we have a network that can have a maximum of 8 devices. The neighbour list will consist of 8 bits. Each bit will be set to One if there is another device in that timeslot. For example, if there is a "neighbour" in timeslot 2, and it is occupying Timeslot 0, then bits 0 and bits 2 will be set to Logic 1 while the rest of the bits are set to 0.

25 1.4.2 Joining a network and adjusting timeslot

When a device is first powered up, it assumes that it is occupying Timeslot #0, and transmits in Timeslot #0. Since it is not in network, and has no neighbours, it will set the TimeSlot parameter in the Transmission packet to 0, and the value of the neighbourhood list is 1 (Bit 0 is set to Logic 1).

30 A device which receives a data packet will process the received data packet as follows. An example algorithm is also shown in Table 1:

- Copy received neighbourhood list

- Copy received time slot
 - If the receiver does not have any neighbour, and it is not in network, it will form a network with the transmitting device.
 - A receiving device adjusts its time slot according to the following rules or predetermined criteria.
- a. If it is not in network, and its ID value is less than transmitter's ID, it will adjust its timeslot
 - b. If it is not in network, but the transmitter is in a network, the receiver will adjust its timeslot.
 - c. If the receiver and the transmitter are already in a network, no timeslot adjustment would be made.

Table 1.

```

if(gbNoNeighbour)
{
    gbNoNeighbour = FALSE;
    if (!gbInNetwork)
    {
        if(gucReceivedID3 > gucMyID3)    Adjust_Timeslot();
        else {
            if(ReceivedPartyInNetwork(guiReceivedNeighList))
                Adjust_Timeslot();
        }
    }
}
gbInNetwork = TRUE; // This is done after adjusting timeslot

```

15

Adjusting Timeslot.

With reference to the flow chart of Figure 7, the algorithm makes timeslot allocation dynamic and automatic without the need for a master device to intervene.

20 The algorithm is as follows:

- Copy Received Timeslot 701.
- Copy received neighbourhood list step 702.
- Search through the received neighbourhood list for an empty timeslot step 703, 704, 705, 706.

- Use the last 3 bits of the ID to start the search step 704. For example, if there are 8 timeslots, and the value of the last 3 bits of the receiver ID is 3, the receiver will first look at slot number 3 in the received neighbourhood list to see if that slot is empty.

5 • If that slot is empty step 706, 707, the receiver's transmit timeslot becomes 3 and the current timeslot of the receiver changed to the received timeslot value.

10 • If that slot is occupied, the receiver will look at the next adjacent timeslot. If the second ID byte is even step 708, it will search in descending order step 709. Otherwise, it will search in ascending order step 710. In this case, if its second ID byte is even, it will look at slot number 2 next.

- This search continues until all 8 slots are exhausted or until an empty timeslot is found.

15 • If an empty timeslot is found step 707, the receiver grabs that slot and it joins the network. It will then operate in NORMAL_MODE thereafter

- If no slots can be found, it will operate in LISTEN_MODE but in NORMAL_MODE.

20 It is also possible to use other search strategies. For example, it is possible to start and proceed in ascending order from timeslot 0, or alternatively in descending order starting from timeslot 12. The inventor has chosen in this preferred embodiment to conduct a search in somewhat random fashion, so that if more than one device detects a network at the
25 same time, the likelihood of both devices taking the same timeslot is diminished.

1.4.3 Network Expansion

In most wireless network protocols, if there are only 8 timeslots, the network can handle only 8 active devices at the most.

30 This algorithm allows the network to expand to cater for more than the number of timeslots available by introducing a Sleep mode to a device in network. Together with Listen mode, it allows devices which were not

previously able to join the network to join in the network. Whenever a device joins a network, it can only remain in the NETWORK_MODE for a certain time period. After this period of time, the device will leave the network and goes into Sleep mode for at least 2 frame periods. When this happens, other
5 devices which are on LISTEN_MODE can grab the time slot and start transmitting in the free slot. Clashes are minimized due to the timeslot allocation algorithm described in 1.4.2.

A device that is in SLEEP MODE will wake up and try to rejoin the network. Since every device will enter into SLEEP_MODE at some point, the
10 network will always have available timeslots.

The State Transitions Diagram in Figure 8 illustrates the entire network link management and access protocol.

Multiple Networks.

The SLEEP MODE feature allows different networks that had been
15 formed to co-exist and eventually merge into one network with all devices synchronized to the same clock.

Suppose 2 different networks had been formed independently and each network are not synchronised. Suppose the networks are mobile and happen to come into each other's vicinity. A device from network A may
20 detect another device in Network B. Since it is already in a network, it will not adjust its timeslot. However, because of the SLEEP MODE feature, it will need to acquire or seek a new network when it wakes up. The process of having a network form up, then break up and then re-form again eventually allows all the devices to get together into one network.

25 1.4.4 Timing Randomisation

The time that a device spends in Normal Mode, Network Mode, Listen Mode, Sleep Mode and Wake Up mode is made random by using the device's IDs and the internal counters in the device. Table 2 shows a code segment of an embodiment in which the method was used.

Table 2

```

#define CNT_SCAN_TIMEOUT      ( 27 + ( (gucMyID3 + gucMyID2) & 0x0F) )
#define CNT_LISTENMODE_TIMEOUT ( 31)
#define CNT_NORMAL_TIMEOUT    ( 81+ ( (gucMyID3 + gucMyID1) & 0x0F) )

// ADD SOME RANDOM TIMING TO WAKE UP TIMER
// TO MAKE IT DIFFERENT FOR EACH UNIT INVOLVE TAG ID
#define CNT_WAKEUP_TIMEOUT    ( ((gucMyID1 ^ TMR0) + (gucMyID3 ^ TMR1L)) & 0x07)
                                |0x08 )
#define CNT_NETWORK_TIMEOUT (61 + ( (gucMyID2 & 0x0F) << 2)

```

2. State Transitions Diagram

Figure 8 shows the State Transitions Diagram that explains the operation of the network protocol.

There are 6 possible states that a device can be in:

- SLEEP 801
- WAKEUP 802
- SCAN 803
- NORMAL 804
- LISTEN 805
- NETWORK 806

In SLEEP state 801, the device is inactive and the transmitter and receiver are powered down. In this state, the device is waiting for external trigger events to wake up the device, such as expiration of timers, external interrupts, user intervention, etc. Once the event triggers, a WAKEUP TIMER is started and the device enters into the WAKEUP state 802.

Once the triggering event occurs, the device powers up and enters into the WAKEUP state 802. In this state, the RF device is powered up. After the RF device is stable, it will transmit a short burst of data containing its ID. The device then starts a timer, known as a SCAN_TIMER, before entering the SCAN state 803. The device will stay in the WAKEUP state 802 until the WAKEUP TIMER expires and the RF device becomes stable.

In SCAN state 803, the device's receiver is turned on continuously. This state is similar to the Listen mode but not Normal Mode as described in

section 2.4. In this state, the device is scanning the air for transmissions from other devices. There are 4 possible scenarios that can happen in this state:

- The device detects transmissions (message) from other devices and is able to grab a time slot for itself. The timeslot is allocated using the algorithm in 1.4.2. The device then starts another timer called the NETWORK_TIMER and then enters the NETWORK state 806.

- The device detects transmissions from other devices but find that there are no time slots available. The device then starts a LISTEN_TIMER and enters the LISTEN state 805.

- 10 • The device could not detect any transmissions (message) and remain in the SCAN state 803 until the SCAN_TIMER expires.

- The SCAN_TIMER expires and the device could not detect other devices. The device is said to have no neighbours. It starts a NORMAL_TIMER and then enters the NORMAL state 804.

15 In the LISTEN state 805, the device has already been synchronized to the timeslots that other devices are using. It is said to be Normal_Mode whereby it is turning on its receiver at the start of a timeslot to listen to transmission from other devices. The transmitter is not turned on since there is no available timeslot.

- 20 • Because of the SLEEP_MODE feature, a time slot will eventually be available and the device will take up that time slot and enters the NETWORK state 806. It will start a NETWORK_TIMER before leaving the LISTEN state 805.

- If no time slot becomes available when the LISTEN_TIMER expires, the device starts a NORMAL_TIMER and leaves the LISTEN state 805 and enters the NORMAL state 806.

25 In the NORMAL state 806, the device will turn on its transmitter and transmit short burst of information including its ID. It will turn on its receiver at the start of every timeslot to listen for messages in the air. In this state, the device could be alone in the network, or it could be in the proximity of other devices but it is not synchronized to other devices, and therefore does not participate or interfere with the other networks. 3 scenarios could happen:

30

• The device detects newly transmitted messages from other devices that it did not previously see. The reason is that other mobile devices could have "walked" into its detection zone. Other possible reason is that another device has detected this device's transmission and adjusts its clock to synchronise to this device. The device then starts a NETWORK TIMER and enters the NETWORK state 806.

• The device did not see any other devices and remains in the NORMAL state 804.

• The NORMAL_TIMER expires and the device enters into the WAKEUP state 802. Before entering the WAKEUP state 802, the device turns off the transmitter and receiver and starts a WAKEUP TIMER.

In the NETWORK state 806, the device will be transmitting and receiving information periodically according to the timeslots. The device will have an allocated timeslot for transmission and reception. It is to be noted that for any device, there may only be one timeslot allocated for transmission and the remaining timeslots are for receiving. The device will remain in this state until one of the following events occurs.

• The NETWORK_TIMER expires. The device then stops all communications, power down the RF device, starts a WAKEUP_TIMER and enters the WAKEUP state 802. This will allow other devices to take over the timeslot that it is occupying so far.

• The number of neighbour devices in the network drops to 0. This could happen in the situation that the device has moved out of the detection range of all other devices (by walking away), or the other devices enter into WAKEUP state 802, and so on. In this case, the device will start a SCAN TIMER and enter into a SCAN state 803.

In time taken for the various timers to expires may vary in different states and in different situations. For example, WAKEUP Timer has a different expiration time when it is started in the SLEEP state 801 and when it is started in the NETWORK state 806.

Applications.

This section lists some of the applications in which the algorithms can be used. However, the algorithms can also be used for other applications.

- Close proximity contact tracing
- Asset tracking
- 5 • Location tracing of people and small devices
- Short range data communication in a small area network
- Security systems.
- RFID related applications

While this invention has been described in connection with specific
10 embodiments thereof, it will be understood that it is capable of further
modification(s). This application is intended to cover any variations uses or
adaptations of the invention following in general, the principles of the invention
and including such departures from the present disclosure as come within
known or customary practice within the art to which the invention pertains and
15 as may be applied to the essential features hereinbefore set forth.

As the present invention may be embodied in several forms without
departing from the spirit of the essential characteristics of the invention; it
should be understood that the above described embodiments are not to limit
the present invention unless otherwise specified, but rather should be
20 construed broadly within the spirit and scope of the invention as defined in the
appended claims. Various modifications and equivalent arrangements are
intended to be included within the spirit and scope of the invention and
appended claims. Therefore, the specific embodiments are to be understood
to be illustrative of the many ways in which the principles of the present
25 invention may be practiced. In the following claims, means-plus-function
clauses are intended to cover structures as performing the defined function
and not only structural equivalents, but also equivalent structures. For
example, although a nail and a screw may not be structural equivalents in that
a nail employs a cylindrical surface to secure wooden parts together, whereas
30 a screw employs a helical surface to secure wooden parts together, in the
environment of fastening wooden parts, a nail and a screw are equivalent
structures. In the example of the preferred embodiments hereinbefore, and in

- particular the data encoding, it is envisaged that some or all of the parameters of the encoding may be altered with respect to each other. For instance, although the START and STOP bits have a duration of half a symbol for each logic level respectively, these durations could be changed to the same duration as a symbol, or may be the duration of one-and-a-half symbol. Nevertheless, in the preferred embodiment described herein, the START and STOP bits have a duration of half a symbol making it easier to differentiate from actual data bits which are one symbol's duration, and also so that valuable air time is not wasted to transmit extra information not being data.
- 10 Accordingly whatever duration is chosen for the START and STOP bits, it may be such that the START symbols and the Preambles will remain never to appear in any of the payload data bytes.

- "Comprises/comprising" when used in this specification is taken to specify the presence of stated features, integers, steps or components but does not preclude the presence or addition of one or more other features, integers, steps, components or groups thereof."